INTRODUCTION

*Computer Systems and C Programming* is a class intended to bring you up to speed on programming small and large programs in C. Originally written in 1978, C remains the most popular programming language, and the most used one in terms of numbers of computer programs written in it.

There are no prerequisites, but you are expected to be at least somewhat familiar with programming and computers (CMPE-12/L is strongly recommended).

In this class, we are going to approach C from an embedded paradigm, and all of your programming assignments are going to be on a 32-bit embedded micro, the MicroChip PIC32.

You will learn how to program in C, how to write modular code, and some of the tips and tricks when dealing with an embedded micro. This is a programming class and you will be writing lots of code. Expect to spend at least 15-20 hours outside of class playing with the code to get things to work.

INSTRUCTOR:

GABRIEL HUGH ELKAIM
E-mail: elkaim@soe.ucsc.edu
Office: Engineering 2 (E2), 337B
Hours: W 2:00-4:00 PM, or by appointment
Phone: (831) 459-3054 (Office)

TAS AND HELPERS:

TA - Bryant Mairs: bwmaids@ucsc.edu E2-316 (ASL), 831-459-2140
Tutors: Jesse Harkin jdharkin@ucsc.edu
Resse Robertson rcrobert@ucsc.edu
Kyle Cordes kjcordes@ucsc.edu
**TEXTBOOKS** (ALSO IN THE LIBRARY ON RESERVE):


---

**Optional Textbooks (good references):**


---

**READINGS**

There is quite a bit of material to cover in this class, and you are expected to have read the assigned reading before coming to class. You will get out of this class what you put into it. Simply put, if you do not do the reading, you will not effectively learn the material. We have gone to quite a bit of trouble in order to find appropriate reading for you, so take the time to read them.

The main textbook, *The C Programming Language*, (K&R) is one of the best books you will ever read on a technical subject. Like the language C itself, it is precise and to the point, and assumes you know what you are doing. There is quite a bit of subtle hints on how to code well in the book, and it should be read carefully. The Notes to Accompany K&R point out some of these
subtleties and expand on some of the examples. Make sure to read both K&R and the accompanying Notes as assigned (you will most likely want to read them again after lecture).

**DEVELOPMENT ENVIRONMENT**

We are going to be using a Microchip 32-Bit Processor, the PIC32, for all of the programming assignments in this class. All of our development will be done within the MPLAB-X integrated development environment (IDE), which can be downloaded from the class website (see the Installing Software handout). We are using the Microchip XC-32 compiler for PIC32. We have put together a small kit that includes a PIC32, and IO_Shield with screen, LEDs, and buttons, and a PicKIT 3 programmer so that you will be able to develop on real hardware. We will also be using the built-in simulator within the MPLAB-X IDE.

All of these tools are available for free, and can be downloaded from the links on the class website. You are strongly encouraged to install these on your own computers so that you can work outside of the lab sections. Note that while these tools are all supposedly cross-platform, we will only be supporting Windows installations.

**GRADING**

This course is based on a combination of the lab and class. They go together and are indivisible (they are divided only because that is how things are done here, ideally it would be just one class). If you cannot complete the lab, you cannot complete the course. You will receive the same grade on the course and the lab, and cannot pass one without passing the other. This is programming class, and you will be graded largely on the programs you write.

- **COURSE/LAB:** 70% Programming assignments
- **30%** In-class quizzes (every week on reading)

Note: there is a sure fire quick and easy way to fail this class—that is to wait until the day before the programming assignment is due to start it. There will be a new programming assignment every week, start it as soon as you get it.

**NOTE ON ATTENDANCE: CMPE-13/L**

There will be a quiz at the beginning of class covering the required reading material at least once a week. No make-up quizzes will be given, nor will there be any opportunities to take the quiz later in that class. If you must miss a quiz, contact the instructor or TAs before the class to make arrangements. The only accepted excuse after a quiz is from the ER.

**WWW SITE, VIDEOS, AND PIAZZA**

Website: [http://classes.soe.ucsc.edu/cmpe013/Winter14/](http://classes.soe.ucsc.edu/cmpe013/Winter14/)

Check this site often as this is where the lecture notes, labs, and lecture videos are posted. You are expected to read the material on the website.
**Videos:** As an experiment in teaching technology, the instructor will be capturing both the audio and screen from the course in real-time. This will later be posted to the website. While every attempt will be made to capture the classes, as this is a new technology, there are no guarantees. Also, while watching the video should be a great way to review course material, if you are not in class you cannot ask questions and/or clarifications. Further, if too few students show up for lectures, the instructor may restrict access to class videos. Note that the class videos will always appear in the Videos subdirectory of the website (even if the link does not work): [http://classes.soe.ucsc.edu/cmpe013/Winter14/Videos/](http://classes.soe.ucsc.edu/cmpe013/Winter14/Videos/)

**Piazza:** [https://piazza.com/ucsc/winter2014/cmpe13l/home](https://piazza.com/ucsc/winter2014/cmpe13l/home)

This term we will be using *Piazza* for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, and the Instructor. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza. If you have any problems or feedback for the developers, email team@piazza.com. Do **NOT** post code or problem solutions unless cleared by the Instructor or TAs (use a private message for that).

---

**COURSE WORK: CMPE 13**

Attendance is highly recommended for the lectures as the material builds up quickly. Lecture material will be made available on the website, usually after covered in class. Annotated lecture notes and videos of the lecture will be posted after class in a timely fashion (see note on video above).

There will be weekly quizzes at the beginning of class that are both required and graded. They are worth 30% of your overall grade, and they are essential to mastering the material. In my experience, understanding the material requires at least three passes—(1) reading the material before lecture, (2) attending lecture and paying attention, and (3) synthesizing the material into your own work during the programming assignments. Skipping any of the steps requires a whole lot more work to make it up on your own.

---

**LAB WORK: CMPE 13 L**

This class should be a lot of fun. I would add also a lot of work (think: drinking from the fire hose). There is a lot to cover, and only 10 weeks to get you familiar and confident in programming in C. Prepare to spend over 20 hours a week on this class, less at first, much more towards the end. We will make every attempt to help you, and to ensure that you succeed, but you have to put in the work yourselves.

You must be enrolled in CMPE 13L to remain in this class. You must pass CMPE 13 to pass CMPE 13L. You will receive the same grade for both the lab and the class, and thus cannot pass one without passing the other. In addition to the class lectures, you are expected to attend lab section at least twice weekly. **If the sections are overcrowded, priority goes to the students signed up for that section;** you are free to go to any additional lab sections as space allows.

We will be working with a 32 bit microcontroller, the MicroChip PIC32, using a development kit manufactured by Digilent (students who wish to buy their own dev kits can purchase them directly from Microchip). These kits will be repurchased from you at the end of the quarter if
they are still in working order. See the section on Development Environment above for details about the tool chain. It is strongly recommended that you install the full development environment on your own computer. You will be working on it quite a bit, and it is nice to be able to work anywhere.

As a programming class, you are going to be doing a lot of programming. The bulk of your grade is based on the programming labs. As such, you will be helping each other out in the labs to understand the material. However, this is NOT license to copy others' work. Credit for collaboration should be explicitly noted; failure to give credit on collaboration is considered a form of cheating and will be dealt with accordingly. Writing code together is strictly forbidden unless explicitly allowed.

**LAB SECTIONS: CMPE 13 L**

Monday, 4:00-6:00PM, Ming Ong: Jesse Harkin and Kyle Cordes  
Tuesday, 7:30-9:30PM, Ming Ong: Bryant Mairs and Kyle Cordes  
Wednesday, 7:00-9:30PM: Ming Ong: Resse Robertson and Jesse Harkin  
Friday, 2:00-4:00PM: JBE 109, Bryant Mairs

There are four lab sections per week starting on Tuesday 07-Jan-2014. You are required to enroll in one, and we will make sure there is space for you in that section. During the lab sections, the tutors and TA will go over tips and tricks, what is required of you, and help you code your assignments. Often, lab assignments require you to show something to the TA or tutors (a check-off), which you will also do in your lab section. You are welcome to any of the other lab sections, but priority is given to those enrolled in that specific lab section.

Note that lab section is when you have access to the TA and tutors, most programming assignments will require you to work on your own outside of those times in order to finish the programming assignment. Make sure you have a set up to do that, and a way to have your files with you (dropbox or your home directories are two excellent choices).

You will be submitting your work using the UNIX submit command to the class course locker, resubmitting a lab will result in overwriting the previous submission.

**ACADEMIC HONESTY**

Academic honesty is a requirement for the course. All assignments must be your own independent work; this includes both quizzes and programming assignments.

What is cheating? It is presenting work that is not yours as your own. You can, and are encouraged to, discuss and strategize with your colleagues on the material and labs, but your work should be your own. Copying is NEVER acceptable.

On the labs, cheating is sharing code unless explicitly told that it is permitted. If a student is caught cheating in either the class or the lab this will result in an immediate failure in the class
and the lab. It will be reported to your college and your department. **DO NOT CHEAT**; it is not worth it.

**ACKNOWLEDGEMENTS**

I would like to acknowledge Cyrus Bazeghi, for all his help with the course material, organization, and lecture material. Steve Summit, who taught an Introductory and Intermediate Programming class at the Experimental College at the University of Washington in Seattle, WA, has generously allowed us to use his notes on K&R and other supplementary materials. Microchip Corp. has generously provided slides and software through their academic partner program, and some of the slides on file systems come from Henry Cheng at UC Davis.